



2183

PATENT

Docket No. JCLA6705

page 1

#2  
Misc.  
letter  
MMA  
4/27/01

In re application of: Shyh-An Chi et al.

Application No.: 09/752,122

Filed: December 29, 2000

For: MEMORY DATA ACCESS STRUCTURE AND  
METHOD SUITABLE FOR USE IN A  
PROCESSOR

Examiner:

Art Unit:

I hereby certify that this correspondence  
and all marked attachments are being  
deposited with the United States Postal  
Service as first class mail in an envelope  
addressed to: Assistant Commissioner  
for Patents, Washington, D.C. 20231, on

March 6, 2001

(Date)

Jiawei Huang, Reg. No. 43,330

ASSISTANT COMMISSIONER FOR PATENTS  
WASHINGTON, D.C. 20231

Sir:

Transmitted herewith is a certified copy of Taiwan Appl. No. 89125861 filed on December 05, 2000. Issue fee related to the above-identified application was paid on the same day as this communication.

A return prepaid postcard is also included herewith.

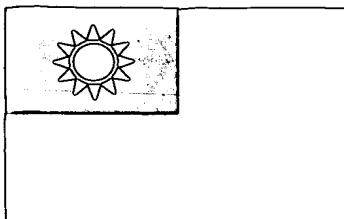
It is believed no fee is due. However, the Commissioner is authorized to charge any fees required, including any fees for additional extension of time, or credit overpayment to Deposit Account No. 50-0710 (Order No. JCLA6705). A duplicate copy of this sheet is enclosed.

  
Jiawei Huang  
Registration No. 43,330

Please send future correspondence to:  
J. C. Patents  
1340 Reynolds Ave., #114  
Irvine, CA 92614  
(949) 660-0761

RECEIVED  
MAR 14 2001  
Technology Center 2100

JCLA6705  
09/752,122



2183

#3  
C. Chang  
5-21-01



# 中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE  
MINISTRY OF ECONOMIC AFFAIRS  
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，  
其申請資料如下：

This is to certify that annexed is a true copy from the records of this  
office of the application as originally filed which is identified hereunder：

申請日：西元 2000 年 12 月 05 日  
Application Date

申請案號：089125861  
Application No.

申請人：智原科技股份有限公司  
Applicant(s)

RECEIVED

MAR 14 2001

Technology Center 2100

局長

Director General

陳明邦

發文日期：西元 2001 年 2 月 1 日  
Issue Date

發文字號：  
Serial No. 09011001151

申請日期	
案 號	
類 別	

A4  
C4

(以上各欄由本局填註)

發明型專利說明書		
一、發明 新型名稱	中 文	適用於處理器之記憶體資料存取架構及其存取方法
	英 文	
二、發明 創作人	姓 名	1 汲世安 2 桂念慈 3 王裕閔
	國 籍	中華民國
	住、居所	1 新竹市光復路一段 531 巷 72-11 號 4 樓 2 台北縣新莊市中原里中信街 87 號 2 樓 3 台中市榮華街 86 號
三、申請人	姓 名 (名稱)	智原科技股份有限公司
	國 籍	中華民國
	住、居所 (事務所)	新竹科學工業園區展業一路 9 號 7 樓之 3
	代 表 人 姓 名	蔡明介

## 四、中文發明摘要(發明之名稱：

適用於處理器之記憶體資料存取  
架構及其存取方法

本發明係提供一種處理器之記憶體資料存取架構及其存取方法，對於每一個由處理器所執行而進入執行階段的指令，其執行結果(Execution Results)將會由處理器所確認(Recognized)，而且經由控制信號傳送到快取記憶體。根據這些控制信號，快取記憶體可決定當所要抓取的指令並未存在快取記憶體時，是否從外部的記憶體抓取此指令。這樣的架構，不論處理器是否具有分支指令預測機制(Branch Prediction Mechanism)皆可不曾使處理器有習知中所產生必須浪費許多的操作時脈，以補償快取記憶體沒有抓到(Miss)的情形，因此，可以顯著地改善整個處理器的效能。。

## 英文發明摘要(發明之名稱：

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

## 五、發明說明 ( | )

本發明是有關一種記憶體資料存取(Memory Data Access)架構及其存取方法，且特別是有關一種適用於一處理器之記憶體資料存取架構及其存取方法。

處理器(Processor)是目前在任何的電子裝置中，皆是不可或缺且廣泛使用的元件。例如，在個人電腦中有中央微處理器(Central Processing Unit)與許多針對不同功能之處理器。而隨著電子裝置的功能日新月異，功能越來越強，其相對地要求處理器所扮演的角色則愈來愈重要，而處理器所需要的功能則也越來越強。

而以往的處理器，對於指令的處理，通常如第 1 圖所示記憶體資料存取(Memory Access)架構之方塊圖，係用以說明處理器與記憶體之間資料存取控制之流程。在此所描述的處理器，則針對中央微處理器為例。此記憶體資料存取架構包括一中央微處理器(CPU)100、一快取記憶體(Cache Memory)120 與一記憶體(Memory)130，其中此中央微處理器 100 係經由一資料匯流排(Data Bus，底下簡稱 DS)102 與此快取記憶體 120 與記憶體 130 相連接，以互相傳送資料。另外，此中央微處理器 100 係經由一位址匯流排(Address Bus，底下簡稱 AB)104 將位址資料傳送到快取記憶體 120 與記憶體 130。而另外，中央微處理器 100 經由一控制信號(Control Signal，CS)106 控制快取記憶體 120。

為方便說明，在此假設中央微處理器 100 的內部分成 3 個管線階段(Pipeline Stage)，亦即在執行指令時，會經過抓取(Fetch)指令、解碼(Decode)指令與執行(Execution)指令

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線

## 五、發明說明(2)

三個階段。首先，中央微處理器 100 會從快取記憶體 120 中抓取指令，接著對於所抓取的指令做解碼的動作，而後在針對所解碼的指令做執行的操作。若是所欲抓取的指令並未在快取記憶體 120 中時，則會從記憶體 130 中抓取所要的指令。而從記憶體 130 中抓取指令的動作中，因為硬體速度上的限制，通常會耗費相當多中央微處理器 100 的運算時脈(Cycles)。

而在中央微處理器 100 所執行的指令中，有一種指令稱為分支指令(Branch Instruction，底下簡稱為 Branch，以利說明)，這種指令係屬於一種控制轉換(Control Transfer instruction)的指令。這是一種要求中央微處理器 100 下一個所要執行的指令係在某一個位址，也就是此中央微處理器 100 必須從目前處理的位址跳到另一位址的指令。而這種指令例如跳躍(Jump)指令、副程式呼叫(Subroutine Call)或是返回(Return)指令等等。

為求說明方便，在此以第 2A 圖所顯示之部分程式片段(Program Segment)作為範例，I 代表中央微處理器 100 所欲執行的指令，而  $I_1$ 、 $I_2$ 、... $I_{10}$ 、 $I_{11}$ 、...等等各分別代表第 1、2、...、10、11...個指令，且指令  $I_1$  為一 Branch 指令，而此指令會在執行  $I_1$  後，跳到第  $I_{10}$  指令。

請參照第 2B 圖，係顯示針對時脈信號與抓取(Fetch)、解碼(Decode)與執行(Execution)三個階段所執行如第 2A 圖所示之程式片段之關係。第 2B 圖所示操作時脈(Clock，此稱為 C)中， $C_1$ 、 $C_2$ 、 $C_3$ ... $C_8$  係分別代表第 1、2、3....、8

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

## 五、發明說明( 3 )

個時脈。當指令  $I_1$  係在執行階段時，也就是在第 3 個時脈  $C_3$  時間點，中央微處理器 100 的抓取單元會開始抓取指令  $I_3$ 。而在此時，若是指令  $I_3$  並未在快取記憶體 120 中時，則必須從記憶體 130 中抓取指令  $I_3$ 。

然而，若是指令  $I_1$  係屬於 Branch 指令，則此指令  $I_1$  將會改變程式所執行的方向，以此例而言，也就是必須開始抓取指令  $I_{10}$ ，然而，此時快取記憶體 120 早已開始將要求抓取指令  $I_3$  的請求送到記憶體 130 中。因此，此時中央微處理器 100 必須等到快取記憶體 120 的抓取指令  $I_3$  請求完成為止。而如第 2B 圖所示，假設從記憶體 130 抓取指令需花費 3 個操作時脈 Cycles 完成，當然，此由於處理器與記憶體之間的速度差距(Gap)越來越大，從記憶體抓取指令所需花費的時脈數目也會隨著越來越多。整個中央微處理器 100 的操作可從第 2B 圖中清楚看出，在 Branch 指令執行之後(時脈  $C_3$  之後)，在第 6 個時脈  $C_6$  之後才開始抓取指令  $I_{10}$ ，浪費了許多的操作時脈。這種情形似乎僅有數個操作時脈的延遲，然而對於高效能且高速處理的處理器而言，這些延遲會對於效能有很大的影響。

習知技藝中有人提出一種具有分支指令預測機制(Branch Prediction Mechanism)，可在抓取階段事先預測指令是否為 Branch 指令及其執行方向(Execution Direction)。然而，上述的問題仍會出現在具有這樣分支指令預測機制的處理器中。假設， $I_1$  是一個跳躍指令(Taken Branch)，會使程式執行方向改變至  $I_{10}$ 。然而，在時脈  $C_1$  抓取到的

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

## 五、發明說明(4)

指令為  $I_1$  時，若是此分支指令預測機制做了錯誤的預測(例如其預測  $I_1$  不是 Branch 指令，或是其預測  $I_1$  並不會改變程式執行方向)，則在進入執行指令  $I_1$  時的時脈  $C_3$  之後，中央微處理器 100 的抓取單元仍會開始抓取指令  $I_3$ ，而若如前述例子中  $I_3$  並未在快取記憶體 120 時，將會產生上述的缺失。反之，若  $I_1$  是 Branch 指令，但其並不會改變程式執行的方向，若是此一分支指令預測機制做了錯誤的預測時，同樣也可能會發生上述的缺失。

有鑑於此，本發明提供一種適用於處理器之記憶體資料存取架構及其存取方法，其可在執行分支指令時，可避免處理器耗費時間抓取目前不會用到的指令，因而不會有操作時脈延遲之情形。

本發明提供一種適用於處理器之記憶體資料存取架構及其存取方法，其可在不論是否具有分支指令預測機制之處理器中，避免執行分支指令時，有浪費操作時脈之情形。

為達上述之目的，本發明提供一種適用於處理器之記憶體資料存取架構，包括一快取記憶體，用以儲存並輸出一指令，其中，該指令係依照一位址信號輸出該指令；以及一管線式處理器，係用以執行複數個處理器指令，該處理器指令至少包括一分支指令，其中該管線式處理器包括一執行單元，根據前一階段傳來的該指令做一執行之操作，並輸出一結果信號與一控制信號，其中該控制信號係用以傳送到該快取記憶體，其中，當該執行單元正在執行該指令為一分支指令時，則該結果信號係為一目標位址，

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線



## 五、發明說明(5)

並經選擇後輸出一位址信號至該快取記憶體，以根據該選擇後之該位址信號抓取下一欲執行的指令，當該執行單元正在執行該分支指令時，該處理器正在對該快取記憶體抓取一抓取指令，而當在執行該分支指令後所得的該控制信號，傳送到該快取記憶體時，若是該抓取指令未在該快取記憶體時，則該快取記憶體將依據該控制信號決定是否對一外部記憶體抓取該抓取指令。

上述之記憶體資料存取架構，更包括一程式計數器，係用以儲存在該些欲執行的指令中，目前所執行之該指令的位址。

上述的記憶體資料存取架構，其中更包括一多工器，用以接收由該執行單元所輸出之該結果信號與該程式計數器所儲存之該執行位址加上一既定值之信號，並選擇輸出其中之一信號成為該位址信號。

為達上述之目的，本發明提供一種適用於處理器之記憶體資料存取架構，包括一快取記憶體，用以儲存並輸出一指令，其中，該指令係依照一位址信號依序輸出該指令；一管線式處理器，係用以執行複數個處理器指令，該些處理器指令至少包括一分支指令，其中該管線式處理器包括一執行單元，並根據前一階段傳來的該指令，做一執行之操作，並輸出一結果信號；一分支指令預測機制，用以根據一抓取指令，輸出一預測位址；一比較器，用以接收該結果信號與該預測位址，並輸出一比較信號，其中當該執行單元正在執行該指令為一分支指令時，則該結果信號係

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線

## 五、發明說明(6)

為一目標位址，並經選擇後輸出一位址信號至該快取記憶體，以根據該位址信號抓取下一欲執行的指令，當該執行單元正在執行該分支指令時，該處理器正在對該快取記憶體抓取一抓取指令，而當在執行該分支指令後所得的該結果信號，將會傳送到該比較器，而比較器會根據該結果信號與該預測位址，比較後輸出該比較信號到該快取記憶體，若是該抓取指令未在該快取記憶體時，則該快取記憶體將依據該比較信號決定是否對一外部記憶體抓取該抓取指令。

如上所述的記憶體資料存取架構，其中更包括一程式計數器，係用以儲存在該些欲執行的指令中，目前所執行之該指令的位址。

上述的記憶體資料存取架構，其中更包括一多工器，用以接收由該執行單元所輸出之該結果信號、該程式計數器所儲存之該執行位址加上一既定值之信號與該預測位址，並選擇輸出其中之一信號成為該位址信號。

為達上述之目的，本發明提供一種適用於處理器之記憶體資料存取方法，包括依照一位址信號依序提供一個指令；執行該指令，並輸出一結果信號與一控制信號，其中，當該所執行之該指令為一分支指令時，則該結果信號係為一目標位址，並經選擇後輸出一位址信號至一快取記憶體，以根據該選擇後之該位址信號抓取下一欲執行的指令，當執行該分支指令時，該處理器將同時抓取一抓取指令，當該抓取指令未在用以儲存該些指令的該快取記憶體

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

## 五、發明說明(17)

時，則該快取記憶體將依據該控制信號決定是否對一外部記憶體抓取該抓取指令。

為達上述之目的，本發明提供一種適用於處理器之記憶體資料存取方法，包括依照一位址信號輸出一指令；執行該指令，並輸出一結果信號；以一分支指令預測機制接收一抓取指令，並輸出一預測位址；比較該結果信號與該預測位址，並輸出一比較信號，其中當所執行的該指令為一分支指令時，則該結果信號係為一目標位址，並經選擇後輸出一位址信號至一快取記憶體，以根據該位址信號抓取下一欲執行的指令，當在執行該分支指令時，該處理器正在抓取一抓取指令，根據該比較信號，若是該抓取指令未在該快取記憶體時，則該快取記憶體將決定是否對一外部記憶體抓取該抓取指令。

為讓本發明之上述目的、特徵、和優點能更明顯易懂，下文特舉一較佳實施例，並配合所附圖式，作詳細說明如下：

圖式之簡單說明：

第 1 圖係繪示習知之記憶體資料存取(Memory Access)架構之方塊圖；

第 2A 圖係繪示作為範例之部分程式片段(Program Segment)；

第 2B 圖係顯示針對時脈信號與三個抓取(Fetch)、解碼(Decode)與執行(Execution)三個階段所執行範例程式片段指令之關係；

## 五、發明說明( 8 )

第 3 圖係顯示本發明一較佳實施例之處理器之記憶體資料存取架構及其存取方法(但不具有分支指令預測機制)；

第 4 圖係顯示本發明另一較佳實施例之處理器之記憶體資料存取架構及其存取方法(但具有分支指令預測機制)；以及

第 5 圖係顯示根據本發明較佳實施例中針對時脈信號與三個抓取(Fetch)、解碼(Decode)與執行(Execution)三個階段所執行範例程式片段指令之關係。

圖式之標號說明：

中央微處理器(CPU)100

快取記憶體(Cache Memory)120

記憶體(Memory)130

一資料匯流排 102

位址匯流排 104

控制信號 106

中央微處理器 300

D 型 Flip-Flop 元件 310、330

解碼器 320

執行單元 340

多工器 350

程式計數器 360

加法器 370

中央微處理器 400

(請先閱讀背面之注意事項再填寫本頁)

裝 · 訂 · 線

## 五、發明說明(9)

D 型 Flip-Flop 元件 410、430、480、481

解碼器 420

執行單元 440

比較器 450

分支指令預測機制 460

多工器 470

### 較佳實施例說明

本發明係提供一種處理器之記憶體資料存取架構及其存取方法，在本發明之架構中，對於每一個由處理器所執行而進入執行階段的指令，其執行結果(Execution Results)將會由處理器所確認(Recognized)，而且經由控制信號傳送到快取記憶體。根據這些控制信號，快取記憶體可決定當所要抓取的指令並未存在快取記憶體時，是否從外部的記憶體抓取此指令。這樣的架構，不論處理器是否具有分支指令預測機制皆不會使處理器有習知中所產生必須浪費許多的操作時脈，以補償快取記憶體沒有抓到(Miss)的情形，因此，可以顯著地改善整個處理器的效能。

請參照第 3 圖，係顯示本發明一較佳實施例之處理器之記憶體資料存取架構及其存取方法。在此架構中，主要係針對不具備分支指令預測機制(Branch Prediction Mechanism)的中央微處理器(CPU)300 說明，然本發明並未限定僅適用於中央微處理器，只要對於必須經由抓取、解碼與執行指令之處理器，都在本發明之範圍。在此假設中央微處理器 300 為一管線式處理器，其內部分成 3 個管線

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線

## 五、發明說明 (10)

階段(Pipeline Stage)，亦即在執行指令時，會經過抓取(Fetch)指令、解碼(Decode)指令與執行(Execution)指令三個階段。

請參照第 3 圖，此中央微處理器 300 包括一 D 型 Flip-Flop 元件 310、一解碼器 320、一 D 型 Flip-Flop 元件 330 與一執行單元 340。D 型 Flip-Flop 元件 310 係接到由快取記憶體 301 經由線路 302 所傳來的指令，之後藉由 D 型 Flip-Flop 元件 312 做一時脈(Clock)上的延遲，並送到解碼器 320。經過解碼器 320 的解碼之後，經由線路 322 傳送解碼後的指令到另一 D 型 Flip-Flop 元件 330 做時脈的延遲，之後藉由線路 332 轉送到執行單元 340 以便執行此指令。

而此執行單元 340 在執行後會將控制信號，例如執行結果(Execution Results)等等傳回快取記憶體 301。而這些執行條件則必須反應出在目前執行階段的指令是否係為分支指令(Branch Instruction)以及其是否跳躍。根據這些控制信號，快取記憶體 301 會決定是否未抓到的指令(Missed Instruction)，也就是目前未儲存在快取記憶體 301 的指令(如本案習知中所介紹的指令  $I_3$ )是否需要抓取。若是不需要，可將不會從外部的記憶體中抓取此指令，也就是不會發出要求以抓取此指令，因此，也不會有如習知技藝中所述，造成時脈上的延誤。

而另外，所執行的結果也會傳回到一多工器 350，若是執行的指令為 Branch 指令，則結果應為一目標位址(Target Address)。而此多工器 350 係連接到此中央微處理

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線

## 五、發明說明(//)

器 300 的程式計數器(Program Counter, 底下簡稱為 PC)360, 此程式計數器 360 係用以儲存在眾多將要執行的指令中, 目前所執行指令的位置。而在多工器 350 與程式計數器 360 中間由一加法器 370。此程式計數器 360 將目前所執行的指令的位置資料傳給加法器 370, 而加法器 370 經過加法運算後傳到多工器 350。若是在執行 Branch 指令之後, 則此 Branch 指令執行的結果與加法器 370 所輸出的資料會經過多工器 350 輸出一位址信號或是目標位址到快取記憶體 301, 以告知下一個欲執行的指令位址。

請參照第 4 圖, 係顯示本發明另一較佳實施例之處理器之記憶體資料存取架構及其存取方法。在此架構中, 主要係針對具備分支指令預測機制(Branch Prediction Mechanism)的中央微處理器(CPU)400 說明, 然本發明並未限定僅適用於中央微處理器, 只要對於必須經由抓取、解碼與執行指令之處理器, 都在本發明之範圍。

請參照第 4 圖, 此中央微處理器 400 包括一 D 型 Flip-Flop 元件 410、一解碼器 420、一 D 型 Flip-Flop 元件 430、一執行單元 440、一比較器 450 與一分支指令預測機制(Branch Prediction Mechanism)460。

D 型 Flip-Flop 元件 410 係接到由快取記憶體 401 經由線路 402 所傳來的指令, 之後藉由 D 型 Flip-Flop 元件 410 做一時脈(Clock)上的延遲, 並送到解碼器 420。經過解碼器 420 的解碼之後, 經由線路 422 傳送解碼後的指令到另一 D 型 Flip-Flop 元件 430 做時脈的延遲, 之後藉由線路 432

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

## 五、發明說明(12)

轉送到執行單元 440 以便執行此指令。

而此執行單元 440 在執行後會將執行結果，與經由本實施例中的分支指令預測機制(Branch Prediction Mechanism)460 根據線路 402 所接收的指令，輸出一預測位址(經過線路 464、一 D 型 Flip-Flop 元件 480、線路 482、D 型 Flip-Flop 元件 481 與線路 483)傳送到比較器 450，比較之後輸出一比較信號，經由線路 452 傳送到快取記憶體 401。而此比較信號即為含有分支指令預測修正的控制信號，而快取記憶體 401 則根據此比較信號決定是否抓取未抓到的指令(Missed Instruction)，也就是目前未儲存在快取記憶體 401 的指令(如本案習知中所介紹的指令  $I_3$ )是否需要抓取。若是不需要，可將不會從外部的記憶體中抓取此指令，也就是不會發出要求以抓取此指令，因此，也不會有如習知技藝中所述，造成時脈上的延誤。

而另外，所執行的結果也會傳回到多工器 470。而此多工器 470 除了接到執行結果外，亦會接到由程式計數器(Program Counter，底下簡稱為 PC)所傳來經過加法器相加的  $PC+4$  之信號 404。除此之外，分支指令預測機制(Branch Prediction Mechanism)460 所輸出的預測位址也會經由線路 462 傳到多工器 470。若是執行單元 440 所執行的指令係 Branch 指令，則其執行結果將會係一目標位址(Target Address)。而根據這些信號，多工器 470 將會傳送一位址信號到快取記憶體 401，以抓取此指令。

請參照第 5 圖，係顯示根據本發明較佳實施例中針對

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線



## 五、發明說明 ( 13 )

時脈信號與三個抓取(Fetch)、解碼(Decode)與執行(Execution)三個階段所執行範例程式片段指令之關係。為清楚地說明與比較，本圖採用與習知之第 2B 圖中相同的情況，以突顯本發明之特中。第 5 圖所示操作時脈(Clock，此稱為 C)中， $C_1$ 、 $C_2$ 、 $C_3 \dots C_8$  係分別代表第 1、2、3...、8 個時脈。當指令  $I_1$  係在執行階段時，也就是在第 3 個時脈  $C_3$  時間點，本實施例中的中央微處理器會到快取記憶體抓取指令  $I_3$ 。而在此時，若是指令  $I_3$  並未在快取記憶體 120 中時，則與習知技藝不同點係根據從中央微處理器所傳來的控制信號(例如執行結果 Execution Results 等等)，會由快取記憶體決定是否到外部的記憶體抓取此指令。

而若是指令  $I_1$  係屬於 Branch 指令，則此指令  $I_1$  將會改變程式所執行的方向，以此例而言，也就是抓取指令  $I_{10}$ ，此時快取記憶體則將同時決定不傳送到外部記憶體要求抓取指令  $I_3$  的請求。因此，此時中央微處理器將在下一時脈開始時抓取 Branch 指令所要執行在目標位址的指令  $I_{10}$ 。如此設計將不必等到快取記憶體的抓取指令  $I_3$  請求完成為止才能繼續抓取目標位址的指令。

依照本發明之處理器之記憶體資料存取架構及其存取方法，則不會浪費了許多的操作時脈。此對於高效能且高速處理的處理器而言，這些所避免的延遲將會對於效能有很顯著的改善。

雖然本發明已以一較佳實施例揭露如上，然其並非用以限定本發明，任何熟習此技藝者，在不脫離本發明之精

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線

## 五、發明說明(14)

神和範圍內，當可作各種之更動與潤飾，因此本發明之保護範圍當視後附之申請專利範圍所界定者為準。

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線

## 六、申請專利範圍

1.一種適用於處理器之記憶體資料存取架構，包括：

一快取記憶體，用以儲存並輸出一指令，其中，該指令係依照一位址信號輸出該指令；以及

一管線式處理器，係用以執行複數個處理器指令，該處理器指令至少包括一分支指令，其中該管線式處理器包括一執行單元，根據前一階段傳來的該指令做一執行之操作，並輸出一結果信號與一控制信號，其中該控制信號係用以傳送到該快取記憶體，其中，

當該執行單元正在執行該指令為一分支指令時，則該結果信號係為一目標位址，並經選擇後輸出一位址信號至該快取記憶體，以根據該選擇後之該位址信號抓取下一欲執行的指令，

當該執行單元正在執行該分支指令時，該處理器正在對該快取記憶體抓取一抓取指令，而當在執行該分支指令後所得的該控制信號，傳送到該快取記憶體時，若是該抓取指令未在該快取記憶體時，則該快取記憶體將依據該控制信號決定是否對一外部記憶體抓取該抓取指令。

2.如申請專利範圍第1項所述之記憶體資料存取架構，其中該控制信號將指示目前在執行階段的指令是否為一個跳躍的分支指令。

3.如申請專利範圍第1項所述之記憶體資料存取架構，其中更包括一程式計數器，係用以儲存在該些欲執行的指令中，目前所執行之該指令的位址。

4.如申請專利範圍第3項所述之記憶體資料存取架構，

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

## 六、申請專利範圍

其中更包括一多工器，用以接收由該執行單元所輸出之該結果信號與該程式計數器所儲存之該執行位址加上一既定值之信號，並選擇輸出其中之一信號成為該位址信號。

5.一種適用於處理器之記憶體資料存取架構，包括：

一快取記憶體，用以儲存並輸出一指令，其中，該指令係依照一位址信號依序輸出該指令；

一管線式處理器，係用以執行複數個處理器指令，該些處理器指令至少包括一分支指令，其中該管線式處理器包括一執行單元，並根據前一階段傳來的該指令，做一執行之操作，並輸出一結果信號；

一分支指令預測機制，用以根據一抓取指令，輸出一預測位址；

一比較器，用以接收該結果信號與該預測位址，並輸出一比較信號，其中

當該執行單元正在執行該指令為一分支指令時，則該結果信號係為一目標位址，並經選擇後輸出一位址信號至該快取記憶體，以根據該位址信號抓取下一欲執行的指令，

當該執行單元正在執行該分支指令時，該處理器正在對該快取記憶體抓取一抓取指令，而當在執行該分支指令後所得的該結果信號，將會傳送到該比較器，而比較器會根據該結果信號與該預測位址，比較後輸出該比較信號到該快取記憶體，若是該抓取指令未在該快取記憶體時，則該快取記憶體將依據該比較信號決定是否對一外部記憶體

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

## 六、申請專利範圍

抓取該抓取指令。

6.如申請專利範圍第5項所述之記憶體資料存取架構，其中更包括一程式計數器，係用以儲存在該些欲執行的指令中，目前所執行之該指令的位址。

7.如申請專利範圍第6項所述之記憶體資料存取架構，其中更包括一多工器，用以接收由該執行單元所輸出之該結果信號、該程式計數器所儲存之該執行位址加上一既定值之信號與該預測位址，並選擇輸出其中之一信號成為該位址信號。

8.如申請專利範圍第5項所述之記憶體資料存取架構，其中該比較信號將指示該分支指令預測機制對於位於執行階段的分支指令的預測是否正確。

9.一種適用於處理器之記憶體資料存取方法，包括：

依照一位址信號依序提供一個指令；

執行該指令，並輸出一結果信號與一控制信號，其中，

當該所執行之該指令為一分支指令時，則該結果信號係為一目標位址，並經選擇後輸出一位址信號至一快取記憶體，以根據該選擇後之該位址信號抓取下一欲執行的指令，

當執行該分支指令時，該處理器將同時抓取一抓取指令，當該抓取指令未在用以儲存該些指令的該快取記憶體時，則該快取記憶體將依據該控制信號決定是否對一外部記憶體抓取該抓取指令。

10.如申請專利範圍第9項所述之記憶體資料存取方

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線

## 六、申請專利範圍

法，其中該控制信號將指示目前在執行階段的指令是否為一個跳躍的分支指令。

11.如申請專利範圍第 9 項所述之記憶體資料存取方法，其中更包括選擇性地輸出該結果信號與該處理器目前正在處理指令之位址加上一既定值其中之一信號。

12.一種適用於處理器之記憶體資料存取方法，包括：  
依照一位址信號輸出一指令；

執行該指令，並輸出一結果信號；

以一分支指令預測機制接收一抓取指令，並輸出一預測位址；

比較該結果信號與該預測位址，並輸出一比較信號，其中

當所執行的該指令為一分支指令時，則該結果信號係為一目標位址，並經選擇後輸出一位址信號至一快取記憶體，以根據該位址信號抓取下一欲執行的指令，

當在執行該分支指令時，該處理器正在抓取一抓取指令，根據該比較信號，若是該抓取指令未在該快取記憶體時，則該快取記憶體將決定是否對一外部記憶體抓取該抓取指令。

13.如申請專利範圍第 12 項所述之記憶體資料存取方法，其中更包括選擇性地輸出該結果信號、該處理器目前正在處理指令之位址加上一既定值、與該預測位址其中之一信號。

14.如申請專利範圍第 12 項所述之記憶體資料存取方

(請先閱讀背面之注意事項再填寫本頁)

裝  
訂  
線

## 六、申請專利範圍

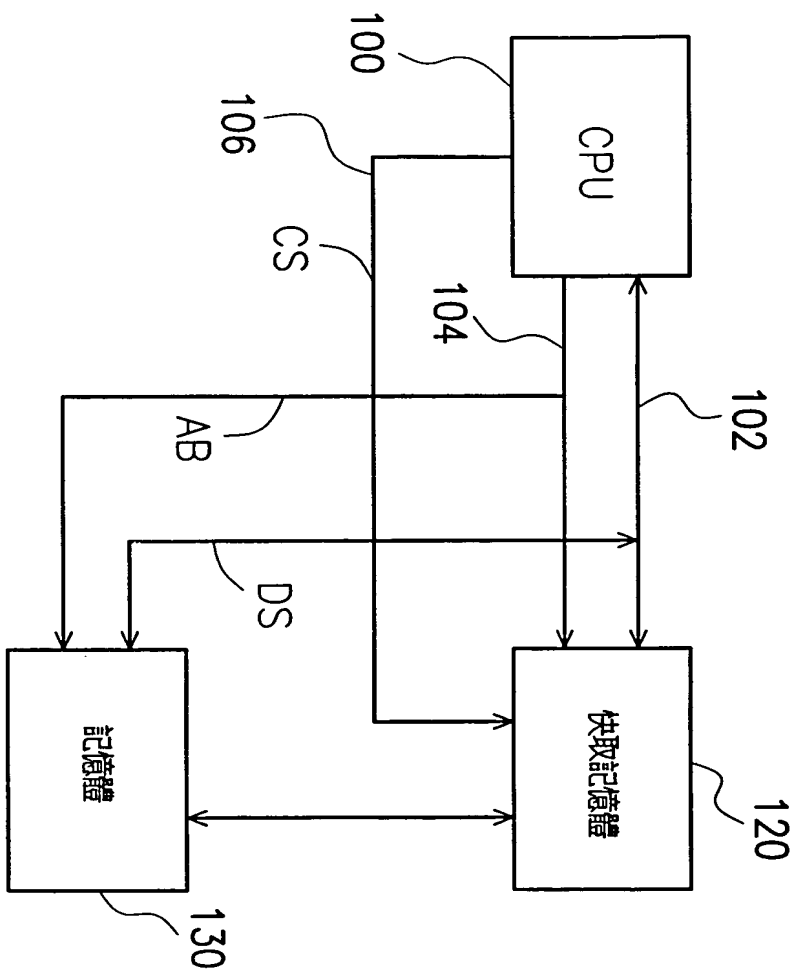
法，其中該比較信號將指示該分支指令預測機制對於位於執行階段的分支指令的預測是否正確。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線



第 1 圖



I1(branch Loop)

I2

I3(未在快取記憶體內)

•

•

Loop I10

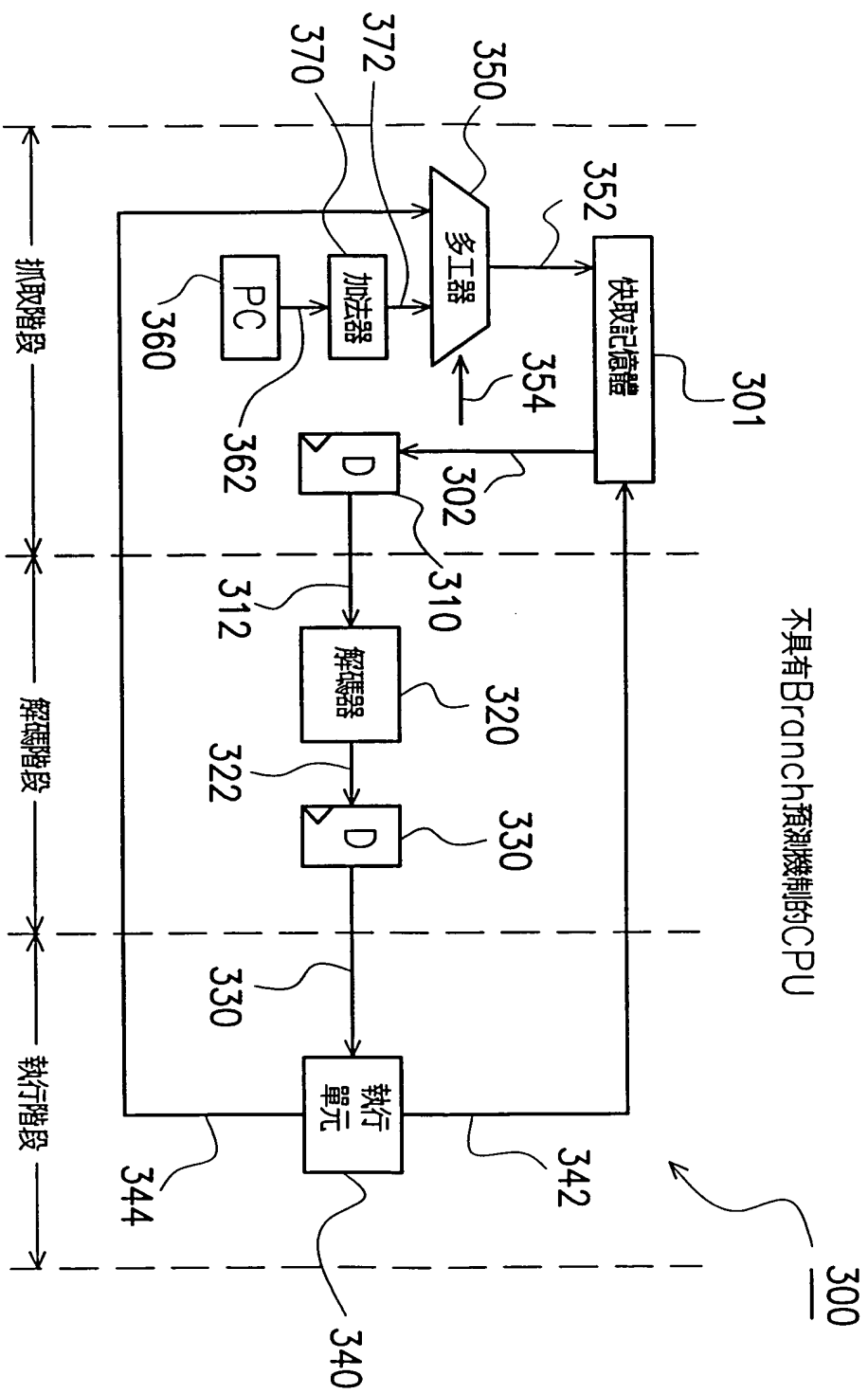
I11

I12

	C1	C2	C3	C4	C5	C6	C7	C8		
抓取	I1	I2	I3	I3	I3	I10	I11	I12		
解碼		I1	I2	I2	I2	I2	I10	I11		
執行			I1	I1	I1	I1	I2	I10		

第2A圖

第2B圖



第 3 圖



	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	
抓取	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>10</sub>	I <sub>11</sub>	I <sub>12</sub>	
解碼		I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>10</sub>	I <sub>11</sub>	
執行			I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>10</sub>	

第 5 圖